

**REMARKS**

Applicant thanks the Examiner for examining the application. Claims 1-4, 6-13, 15-22, 24-31, and 33-44 are now pending.

*Claim Rejections – 35 U.S.C. § 102(e)*

The Examiner rejected claim 6 under 35 U.S.C. § 102(e) as being clearly anticipated by U.S. Patent No. 6,477,586 to Achenson et al.

Applicant's independent claim 6 requires, among other things, in a worker thread not associated with the assigned task queue, processing the task. The Examiner cited to col. 5 lines 42-45 and 60-63 and col. 6 line 64 to col. 7 line 9 of Achenson et al. as teaching or suggesting this limitation. The Examiner makes the exact same rejection of the corresponding limitation found in Applicant's independent claim 1 (among others).

However, when responding to Applicant's arguments regarding Achenson et al. and this particular limitation, the Examiner states "Thus, when a message is transfer [sic] to process 3 from process 2A as states in col. 5 lines 60-63 and col. 6 line 64 – col. 7 line 9, it [Achenson et al.] **suggests and/or indicates** that a worker thread within process 3 is processing the message/task from a queue of process 2A." Office Action page 8, element 19 (emphasis added).

With all due respect to the Examiner, and leaving aside the merits of that argument for the moment, a rejection under § 102 **cannot** rely on suggestion or implication, but rather must rely on what is actually disclosed. The Examiner herself is arguing, in response to Applicant's arguments, that Achenson et al. does not disclose the limitation of "in a worker thread not associated with the assigned task queue, processing the task", but rather suggests it. Thus, the Examiner **cannot** maintain the rejection of Applicant's independent claim 6 under § 102(e), because the reference does not actually disclose all of the limitations of Applicant's independent claim 6. This rejection is therefore improper, and Applicant respectfully submits that if any rejection based on this reference is to be maintained on Applicant's independent claim 6, that rejection should be based on § 103.

Applicant will now address the substance of the Examiner's argument, which will also apply to Applicant's independent claim 6, below with regards to Applicant's independent claim 1.

*Claim Rejections – 35 U.S.C. § 103(a)*

The Examiner rejected claims 1, 4, 7, 10, 13, 15-16, 19, 22, 24-25, 28, 31, 33, 36, 39, and 40 under 35 U.S.C. § 103(a) as being unpatentable over Achenson et al., as applied to claim 6 above, in view of U.S. Patent No. 5,438,680 to Sullivan.

Applicant's independent claim 1 requires, among other things, from a worker thread, processing a task from a task queue not associated with the thread. The Examiner cited to col. 5 lines 42-45 and 60-63 and col. 6 line 64 to col. 7 line 9 of Achenson et al. as teaching or suggesting this limitation. Further, as stated above, the Examiner makes the following argument regarding Achenson et al.: "Thus, when a message is transfer [sic] to process 3 from process 2A as states in col. 5 lines 60-63 and col. 6 line 64 – col. 7 line 9, it [Achenson et al.] suggests and/or indicates that a worker thread within process 3 is processing the message/task from a queue of process 2A." Office Action page 8, element 19.

However, nowhere in the cited text, nor in any other text, does Achenson et al. teach or suggest from a worker thread, processing a task from a task queue not associated with the thread, as required by Applicant's independent claim 1. Indeed, Applicant respectfully submits that the Examiner's claim that Achenson et al. "suggests and/or indicates that a worker thread within process 3 is processing the message/task from a queue of process 2A" has no basis in Achenson et al.

The cited text of Achenson et al. states as follows:

Similarly the thread and queue pairs shown in blocks 46, 48, 50, 52, 54, 56, 58, 60 and 62 are worker threads and queues in the illustrated example of FIG. 2

. . .

Alternatively, the worker thread receiving the RPC message may indicate that message is not in the appropriate process within the distributed system to handle the RPC request and that the RPC message is to be forwarded to another process.

...

In a case where an RPC request message is determined by the worker thread of Process 2A to be one which is not processable by that worker thread, the worker thread will determine from data stored in process 2A which other process in the distributed system will be able to deal with the RPC request. The RPC request message is then placed on the appropriate queue for transfer to a further process. For example, the RPC message in process 2A of block 32 can be placed in queue Qc shown in block 78 for transfer to process 3 of block 34 via connection 92. The message thread, or associated send function block, sends the RPC request message over connection 92 to process 3 where message thread in block 80 receives the message.

The Examiner argues that this text suggests that some thread in process 3 is working on a task that is queued in a queue that is part of process 2A. Applicant respectfully fails to see how the Examiner arrives at this conclusion based on the cited text reproduced above. The cited text says that a message in process 2A “can be placed in queue Qc shown in block 78” (which is part of process 2A) “for transfer to process 3” and that the message from queue Qc is then sent “over connection 92 to process 3” where a thread “receives the message”. This text clearly teaches that, instead of the message remaining in the queue Qc in process 2A, the message is transferred to process 3. Indeed, Achenson et al. goes on to state in lines 10-11 of col. 7 that “Before the RPC message **is forwarded** by process 2A” and then again in lines 21-22 “The RPC message which **is communicated** from process 2A to process 3”. All of this teaches that the message does not remain in a queue of process 2A, as the Examiner implies, but rather goes to a queue of process 3 – where process 3 then acts on the message (“Process 3 can deal with the message in the same manner as process 2A dealt with the message”, col. 7 lines 23-25). Indeed, nothing in Achenson et al. even seems to imply that the message remains in a queue of process 2A. All of the available evidence in Achenson et al. says the message goes on to process 3, contrary to the Examiner’s argument.

What Achenson et al. teaches is in direct contrast to Applicant’s independent claim 1, which requires that, from a worker thread, processing a task from a task queue not associated with the thread. Therefore, Achenson et al. fails to teach or suggest this

limitation of Applicant's independent claim 1, and thus Achenson et al. does not teach or suggest Applicant's independent claim 1. Applicant's independent claim 1 is therefore allowable over Achenson et al., and is further allowable over the combination of Achenson et al. with Sullivan.

Applicant's independent claims 6, 10, 15, 19, 24, 28, 33, 39, and 40 all include limitations similar to those of Applicant's allowable amended independent claim 1. Therefore, for at least the reason(s) given above with regards to Applicant's allowable independent claim 1, Applicant's independent claims 6, 10, 15, 19, 24, 28, 33, 39, and 40 are themselves not taught or suggested by Achenson et al. in view of Sullivan, and thus, Applicant's independent claims 6, 10, 15, 19, 24, 28, 33, 39, and 40 are themselves allowable over the combination of Achenson et al. with Sullivan.

Applicant's dependent claims 4, 7, 13, 16, 22, 25, 31, and 36 depend from, respectively, Applicant's allowable independent claims 1, 6, 10, 15, 19, 24, 28, and 33. Therefore, for at least the reasons given above with regards to Applicant's allowable independent claims 1, 6, 10, 15, 19, 24, 28, and 33, Applicant's dependent claims 4, 7, 13, 16, 22, 25, 31, and 36 are themselves not taught or suggested by Achenson et al. in view of Sullivan, and thus, Applicant's dependent claims 4, 7, 13, 16, 22, 25, 31, and 36 are themselves allowable over the combination of Achenson et al. with Sullivan.

The Examiner then rejected claims 2, 3, 8, 9, 11, 12, 17, 18, 20, 21, 26, 27, 29, 30, 34, and 35 under 35 U.S.C. § 103(a) as being unpatentable over Achenson et al. in view of Sullivan, as applied to claims 1, 6, 10, 15, 19, 24, 28, and 33 above, and further in view of U.S. Patent No. 6,377,984 to Najork et al.

Applicant's dependent claims 2, 3, 8, 9, 11, 12, 17, 18, 20, 21, 26, 27, 29, 30, 34, and 35 depend from, respectively, Applicant's allowable independent claims 1, 6, 10, 15, 19, 24, 28, and 33. Therefore, for at least the reasons given above with regards to Applicant's allowable independent claims 1, 6, 10, 15, 19, 24, 28, and 33, Applicant's dependent claims 2, 3, 8, 9, 11, 12, 17, 18, 20, 21, 26, 27, 29, 30, 34, and 35 are themselves not taught or suggested by Achenson et al. in view of Sullivan, and thus,

Applicant's dependent claims 2, 3, 8, 9, 11, 12, 17, 18, 20, 21, 26, 27, 29, 30, 34, and 35 are also not taught or suggested by Achenson et al. in view of Sullivan and further in view of Najork et al. For these reasons alone, Applicant's dependent claims 2, 3, 8, 9, 11, 12, 17, 18, 20, 21, 26, 27, 29, 30, 34, and 35 are themselves allowable over the combination of Achenson et al. with Sullivan and further with Najork et al.

Applicant further notes that Applicant's dependent claim 3 is, on its face, itself not taught or suggested by Najork et al. Applicant's dependent claim 3 requires that selecting comprises **determining whether a selected task queue is in a busy state** (emphasis added). The Examiner cited to col. 3 lines 22-33 of Najork et al. as teaching or suggesting this limitation. Further, the Examiner argues, "**When Najork teaches of selecting an empty task queue**, he inherently teaches the step of determining whether the selected queue is in a busy state in order to determine that the queue is empty. If a queue is in a busy state, it would not be selected and/or considered as an empty queue." Office action, page 8, section 20 (emphasis added).

The Examiner's argument, however, does not properly characterize what the cited text of Najork et al. teaches. The cited text of Najork et al. does not teach **selecting** an empty task queue. The cited text of Najork et al. does not even *suggest selecting* an empty task queue. Rather, what the cited text of Najork et al. actually teaches in the cited paragraph is that **when an underlying queue becomes empty**, certain actions may be taken by the thread correspond to that queue; *see at least* Najork et al. col. 3 lines 25-31 (emphasis added). **None of those** actions include the selection of an empty task queue, or a busy queue, or the consideration of a task queue as being empty or busy. The cited text of Najork et al. merely states that **when** a queue becomes empty, something happens. Arguably Najork et al. must first determine that the queue is empty, but the cited text of Najork et al. offers no guidance whatsoever on **how** the determination that the queue is empty is made. Indeed, Applicant can think of a variety of ways in which Najork et al. may determine that a queue is empty; none of these were even remotely suggested or implicated by the cited text of Najork et al. Indeed, the cited text of Najork et al. does not offer **anything** along these lines – particularly not that a queue is determined to be empty by first determining that it is in a

busy state, and then negatively inferring that a busy state means a queue is not empty, as the Examiner argues. With all due respect to the Examiner, that suggestion comes from the Examiner, not the cited text of Najork et al., and the cited text of Najork et al., which merely describes **what happens when a thread is empty**, cannot be stretched to make that argument without breaking.

Nothing in the cited text of Najork et al. says anything about determining whether a selected task queue is in a busy state, as required by Applicant's dependent claim 3. Thus, for this reason alone, Najork et al. does not teach or suggest Applicant's dependent claim 3, and therefore Applicant's dependent claim 3 is itself allowable over Najork et al. and is also allowable over the combination of Achenson et al. and Sullivan with Najork et al.

Applicant's dependent claims 9, 12, 18, 21, 27, 30, and 35 all contain limitations similar to those of Applicant's allowable dependent claim 3. Therefore, for at least the reason given above with regards to Applicant's allowable dependent claim 3, Najork et al. does not teach or suggest Applicant's dependent claims 9, 12, 18, 21, 27, 30, and 35, and therefore Applicant's dependent claims 9, 12, 18, 21, 27, 30, and 35 are themselves allowable over Najork et al. and are also allowable over the combination of Achenson et al. and Sullivan with Najork et al.

The Examiner finally rejected claims 37, 38, and 41-44 under 35 U.S.C. § 103(a) as being unpatentable over Achenson et al. in view of Sullivan and further in view of U.S. Published Patent App. No. 2003/0225815 to Brenner et al.

Applicant's independent claim 37 requires, among other things, assigning a task to a task queue in an essentially random fashion, comprising: using a random number generator to identify an initial task queue; upon determining that the initial task queue is not empty, searching the other task queues for an empty queue; and upon finding an empty task queue, storing the task in the empty task queue. The Examiner cites to paragraph 0043, paragraph 0095, and Fig. 8 of Brenner et al. as teaching or suggesting upon determining that the initial task queue is not empty, searching the other task queues for an empty queue, and upon finding an empty task queue, storing the task in

the empty task queue.

The cited text of Brenner et al. states as follows:

[0043] As shown in FIG. 2, when an unbound new thread Th13 is created as part of a new process, or job, the dispatcher 150 attempts to place the thread in a run queue associated with an idle CPU. To do this, the dispatcher 150 performs a round-robin search among the CPUs 230-280 of the system 200. If an idle CPU is found, the new thread Th13 is added to the local run queue of the idle CPU.

[0095] If the new thread is part of an existing process (step 840: YES), the initial load balancing device 730 performs a round robin search of the CPUs of the node to which the other threads from the existing process were assigned (step 850) looking for an idle CPU. If the new thread is not part of an existing process (step 840: NO), the initial load balancing device 730 performs a round robin search of all nodes and CPUs for an idle CPU (step 860).

In other words, Brenner et al. is looking for an idle CPU, and it uses some type of round-robin search to try to find one. As can be seen from the text, Brenner et al. offers no details as to the particulars of the search. However, the Examiner argues that when Brenner et al. "search through and/or scan the nodes, he inherently teaches the step of determining whether the initial task queue is empty or not before he can determine that the next processor is idle (not busy) which implies its queue is empty." Office Action page 9 element 22. But as shown in the text above, Brenner et al. does not say this. Brenner et al. simply states that a round-robin search among a number of CPUs is performed to attempt to find an idle CPU. Brenner et al. says nothing, and indeed implies nothing whatsoever, about **how** that search is performed. Applicant can think of a variety of ways in which Brenner et al. might perform the search for an idle CPU, including the Examiner's suggestion. However, Brenner et al. itself **does not suggest or even imply how the search is actually performed. All Brenner et al. teaches is that a search is performed.** With all due respect to the Examiner, what the Examiner states that Brenner et al. "inherently teaches" rather seems to be hindsight on the part of the Examiner, because there is no basis in Brenner et al. for the Examiner's assertion about the round-robin search for an idle CPU is performed.

Thus, Brenner et al. does not teach or suggest all the limitations of Applicant's independent claim 37, and therefore, the combination of Achenson et al. with Sullivan and further with Brenner et al. does not teach or suggest Applicant's independent claim

37. Applicant's independent claim 37 is therefore allowable over Brenner et al. and is also thus allowable over the combination of Achenson et al. with Sullivan and further with Brenner et al.

Applicant's independent claims 38 and 41 include limitations similar to those of Applicant's allowable independent claim 37. Therefore, for at least the reason given above with regards to Applicant's allowable independent claim 37, Applicant's independent claims 38 and 41 are not taught or suggested by Brenner et al. and thus are not taught or suggested by the combination of Achenson et al. with Sullivan and further with Brenner et al. Applicant's independent claims 38 and 41 are therefore themselves allowable over the combination of Achenson et al. with Sullivan and further with Brenner et al.

Applicant's dependent claims 42-44 depend from, respectively, Applicant's allowable independent claims 37, 38, and 41. Therefore, for at least the reasons given above with regards to Applicant's allowable independent claims 37, 38, and 41, Applicant's dependent claims 42-44 are themselves not taught or suggested by Brenner et al., and thus, Applicant's dependent claims 42-44 are themselves allowable over the combination of Achenson et al. with Sullivan and with Brenner et al.

### **CONCLUSION**

Applicant believes this Amendment and Response to be fully responsive to the present Office Action. Thus, based on the foregoing Remarks, Applicant respectfully submits that this application is in condition for allowance. Accordingly, Applicant requests allowance of the application.

Applicant hereby petitions for any extension of time required to maintain the pendency of this case. If there is any fee occasioned by this response that is not paid, please charge any deficiency to Deposit Account No. 50-3735.

Should the enclosed papers or fees be considered incomplete, Applicant respectfully requests that the Patent Office contact the undersigned collect at the telephone number provided below.



Applicant invites the Examiner to contact the Applicant's undersigned Attorney if any issues are deemed to remain prior to allowance.

Respectfully submitted,

\_\_\_\_\_  
/SPM/

Shaun P. Montana, Esq.  
Attorney for Applicant(s)  
Registration No.: 54,320  
Chapin Intellectual Property Law, LLC  
Westborough Office Park  
1700 West Park Drive  
Westborough, Massachusetts 01581  
Telephone: (508) 616-9660  
Facsimile: (508) 616-9661

Attorney Docket No.: OID06-31(16101)

Dated: September 10, 2007